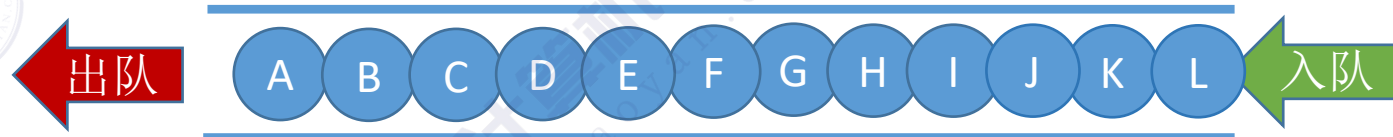
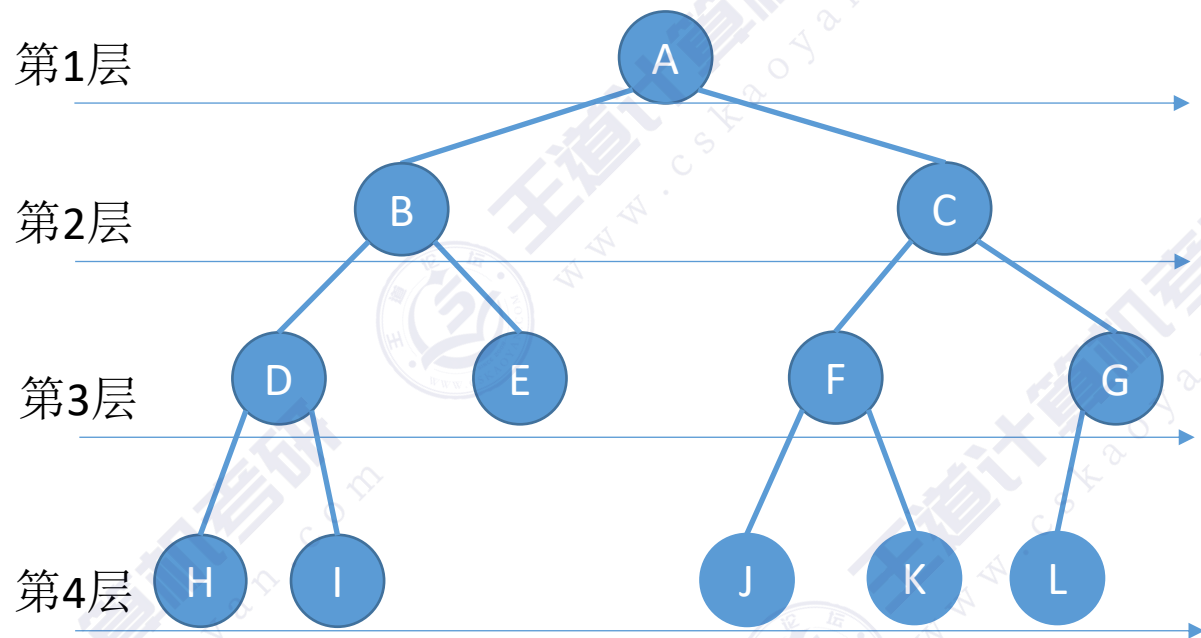


本节内容

二叉树

层序遍历

二叉树的层序遍历



算法思想:

- ①初始化一个辅助**队列**
- ②根结点入队
- ③若队列非空, 则队头结点出队, 访问该结点, 并将其左、右孩子插入队尾 (如果有的话)
- ④重复③直至队列为空

代码实现

算法思想:

- ①初始化一个辅助**队列**
- ②根结点入队
- ③若队列非空, 则队头结点出队, 访问该结点, 并将其左、右孩子插入队尾 (如果有的话)
- ④重复③直至队列为空

//层序遍历

```
void LevelOrder(BiTree T){
```

```
    LinkQueue Q;
```

```
    InitQueue(Q);
```

```
    BiTree p;
```

```
    EnQueue(Q,T);
```

```
    while(!IsEmpty(Q)){
```

```
        DeQueue(Q, p);
```

```
        visit(p);
```

```
        if(p->lchild!=NULL)
```

```
            EnQueue(Q,p->lchild);
```

```
        if(p->rchild!=NULL)
```

```
            EnQueue(Q,p->rchild);
```

```
    }
```

```
}
```

//初始化辅助队列

//将根结点入队

//队列不空则循环

//队头结点出队

//访问出队结点

//左孩子入队

//右孩子入队

//二叉树的结点 (链式存储)

```
typedef struct BiTNode{
```

```
    char data;
```

```
    struct BiTNode *lchild,*rchild;
```

```
}BiTNode,*BiTree;
```

//链式队列结点

```
typedef struct LinkNode{
```

```
    BiTNode * data;
```

```
    struct LinkNode *next;
```

```
}LinkNode;
```

```
typedef struct{
```

```
    LinkNode *front,*rear; //队头队尾
```

```
}LinkQueue;
```

存指针而
不是结点

知识回顾与重要考点

树的层次遍历算法思想:

- ①初始化一个辅助**队列**
- ②根结点入队
- ③若队列非空，则队头结点出队，访问该结点，并将其左、右孩子插入队尾（如果有的话）
- ④重复③直至队列为空